آزمون پایان نیم‌سال درس "طراحی و پیاده‌سازی زبان‌های برنامه‌سازی"

تاریخ: ۱۴۰۰/۴/۸ زمان آزمون: ۷۵+۷۵ دقیقه

نکات مهم:

۱- **امتحان شامل دو بخش ۷۵ دقیقه ای است. تا قبل از پایان ۷۵ دقیقه اول بایستی جواب بخش اول آپلود شود و سپس ۷۵ دقیقه دوم برای پاسخ به بخش دوم. در کوئرا هر بخش جداگانه در نظر گرفته شده است.**

۲- جواب سوالهای هر بخش را به صورت دست نوشته بنویسید. سپس جواب ها را به ترتیب شماره سوالات در یک فایل pdf با نام شماره دانشجوییتان گذاشته و در قسمت مربوطه در کوئرا آپلود کنید. دقت کنید که در کوئرا تنها امکان ارسال pdf وجود دارد. تنها ارسال نهایی شما در کوئرا تصحیح خواهد شد. دقت کنید که در صورت ارسال چند جواب، کوئرا به صورت خودکار آخرین ارسال را به عنوان ارسال نهایی در نظر می گیرد. برای رعایت عدالت و نظم، در زمان برگزاری آزمون به پرسشی پاسخ داده نخواهد شد.

۳- استفاده از هر منبع مکتوب یا اینترنتی در دسترس شخص دانشجو مجاز است. اما مشورت با دیگری (هر کسی که باشد و در هر اندازه) مجاز نیست. لطفا جواب سوال ها را به دیگران منتقل نکنید و از پاسخ دیگران حتی اگر در دسترس شما قرار گرفت استفاده یا کپی نکنید. در صورت وقوع چنین مواردی مطابق با آیین نامه های دانشگاه رفتار خواهد شود.

موفق، سلامت و پیروز باشید.

ایزدی

بخش اول (۷۵ دقیقه)

۱- تابع SQ را به شکل زیر تعریف می‌کنیم:

$$SQ((f_1, f_2, \dots, f_k), x) = f_k(f_{k-1} \dots (f_2(f_1(x))$$

این تابع SQ دو آرگومان ورودی دارد: آرگومان اول لیستی از تابعهای یک آرگومانی و آرگومان دوم متغیر x است و خروجی آن مساوی به کارگیری متوالی هر یک از توابع ورودی بر خروجی تابع قبلی است. به مثال زیر توجه کنید:

```
(SQ (list (lambda (x) (+ x 1)) (lambda (x) (* x 3)) (lambda (x) (- 100 x))) 4)
```

خروجی این برنامه عدد ۸۵ است که از محاسبه عبارت ((3 (1 4 +) *) 100 -) حاصل شده است.

الف- در زبان راکت پیاده سازی تابع SQ را به روش بازگشتی صریح بنویسید. (یعنی تابع پیاده سازی شده بایستی خودش بازگشتی باشد و هیچ تابع کمکی تعریف نکنید و از توابع مرتبه دوم یعنی توابعی که آرگومان ورودی از نوع تابع می‌پذیرند مانند map ، filter ، foldl و ... استفاده نکنید.)

ب- این بار برعکس، تابع SQ را به طور غیربازگشتی و با استفاده از توابع مرتبه دومی که بر لیستها عمل می‌کنند پیاده سازی کنید.

۲- می‌خواهیم زبان CHECKED (معرفی شده در فصل ۷ کتاب) را به گونه ای توسعه دهیم که مانند زبان C هر مقدار از نوع عدد صحیح نقش مقدار از نوع بولین نیز داشته باشد به طوری که هر جایی از یک عبارت، مقداری از نوع بولین مورد نیاز است مقدار عدد صحیح نیز مقبول است: صفر نقش false و هر عدد غیر صفر نقش true دارد. مقادیر بولین هم همچنان وجود دارند و قابل استفاده هستند. تغییرات لازم در تعریف نوع ها، ساختارهای داده ای انواع و پیاده سازی تابع type-of را بیان کنید.

۳- حاصل (خروجی) برنامه‌های زیر را تعیین نمایید و با الهام از آنها تفاوت lexical scoping و dynamic scoping را توضیح دهید.

Scheme: (lexical scoping)	Common Lisp: (dynamic scoping)
(define x 3) (define (f y) (+ y x)) (define (g x) (f 100)) (g 4)	(defvar x 3) (defun f (y) (+ y x)) (defun g (x) (f 100)) (g 4)

بخش دوم (۷۵ دقیقه)

۴- می دانیم که در زبان IMPLICIT-REFS (تعریف شده در فصل ۴ کتاب) نحوه تعریف یک تابع بر اساس ساختار نحوی زیر است:

Expression ::= proc (*Identifier*) *Expression*

و نحوه فراخوانی پارامترها در این مورد، فراخوانی با مقدار (Call-by-Value) است، یعنی اگر f تابعی از این تعریف باشد هرگاه f(x) فراخوانی شود مقدار x به تابع ارسال می‌شود.

الف- آقای پاسکال (Pascal)، ساختار نحوی زیر را به زبان فوق اضافه نموده است:

Expression ::= procvar (*Identifier*) *Expression*

که نحوه فراخوانی پارامترها در این ساختار، فراخوانی با رفرنس (Call-by-Reference) خواهد بود، یعنی اگر f تابعی از این تعریف باشد هرگاه f(x) فراخوانی شود آدرس x به تابع ارسال می‌شود. .

زبان IMPLICIT-REFS به اضافه ساختار procvar را **زبان Pascal** می نامیم.

ب- خانم ادا (Ada)، نه تنها ساختار procvar پیشنهادی پاسکال را قبول ندارد، بلکه ساختار proc در خود زبان IMPLICIT-REFS را نیز ساختار مناسبی نمی‌داند. لذا پیشنهاد می‌کند به جای proc از سه ساختار نحوی زیر استفاده کنیم:

Expression ::= proc-in (*Identifier*) *Expression*
Expression ::= proc-out (*Identifier*) *Expression*
Expression ::= proc-inout (*Identifier*) *Expression*

از نظر خانم ادا، ما به عنوان طراح زبان برنامه سازی نباید تعیین چگونگی فراخوانی پارامتر توابع را در اختیار برنامه نویس قرار دهیم. بلکه فقط باید از برنامه نویس بخواهیم وقتی پارامتری برای یک تابع در نظر می‌گیرد تعیین کند در هنگام فراخوانی تابع آیا می‌خواهد از پارامتر به عنوان یک آورنده مقدار داده (ورودی)، یا یک دریافت کننده نتیجه تابع (خروجی) یا در هر دو نقش ورودی و خروجی استفاده کند. بنابر این:

- اگر f تابعی از ساختار proc-in باشد هرگاه f(x) فراخوانی شود، اجرای تابع f فقط از مقدار x استفاده می‌کند و نمی‌تواند آن را تغییر دهد.
- اگر f تابعی از ساختار proc-out باشد هرگاه f(x) فراخوانی شود، در زمان اجرای تابع f از مقدار x استفاده‌ای نمی‌شود اما خروجی تابع در متغیر x قرار می‌گیرد.
- اگر f تابعی از ساختار proc-inout باشد هرگاه f(x) فراخوانی شود،در اجرای تابع f هم از مقدار x استفاده می‌شود و هم خروجی تابع در متغیر x قرار می‌گیرد.

از نظر خانم ادا، این که در زمان اجرا، یک پارامتر فراخوانی با مقدار یا فراخوانی با رفرنس شود تصمیمی است که پیاده ساز زبان (یعنی کامپایلر نویس یا طراح مفسر) می‌گیرد و در درون کامپایلر یا مفسر پیاده‌سازی می‌کند. زبان IMPLICIT-REFS که به جای ساختار proc از سه ساختار بالا استفاده می‌کند را **زبان Ada** می‌نامیم.

نکته: در مباحث بالا و سوالات زیر، فرض می کنیم که مقدار متغیرها و پارامترها می‌توانند از انواع داده ساده مانند اعداد صحیح یا بولین یا از انواع داده پیچیده مانند لیست یا آرایه باشند.

اکنون به سوالات زیر پاسخ دهید:

۴-۱. به نظر شما دلایل خانم ادا برای ترجیح پیشنهاد خود به دو روش دیگر به خصوص ارجحیت روش او به روش پاسکال در چیست؟

۴-۲. (در مقایسه با زبان ادا) چه مزیتها یا معایبی در زبان پاسکال وجود دارد؟ (هم از نگاه برنامه نویس و هم از نگاه پیاده ساز زبان.)

۴-۳. زبان ادا در سطح بالاتری است یا زبان پاسکال؟ (ابتدا بگویید تعریف شما از سطح بالا یا پایین بودن یک زبان در مقایسه با زبان دیگر چیست؟)

۴-۴. اگر شما بخواهید سه ساختار پیشنهادی خانم ادا را پیاده سازی کنید، از کدام روش فراخوانی پارامترها استفاده می‌کنید؟ به عبارت دیگر، در چه شرایطی برای پیاده سازی ساختارهای ادا از فراخوانی با مقدار و در چه شرایطی از فراخوانی با رفرنس استفاده می‌کنید؟

۴-۵. (با فرض این که جواب پرسش قبلی شما درست است) برای پیاده سازی زبان ادا، تغییرات لازم در تعاریف و قوانین و پیاده سازی توابع مفسر را بیان کنید. (پیاده سازی زبان IMPLICIT-REFS که در بخش ۳-۴ و ۴-۵ کتاب ارایه شده است را پیش فرض بگیرید و فقط تغییرات لازم روی آنها برای پیاده سازی زبان ادا را بنویسید.)

پایان

موفق باشید